Identifying Mis-Configured Author Profiles on Google Scholar Using Deep Learning

Tang, Jiaxin

2021-08

Tang , J , Chen , Y , She , G , Xu , Y , Sha , K , Wang , X , Wang , Y , Zhang , Z & Hui , P 2021 , ' Identifying Mis-Configured Author Profiles on Google Scholar Using Deep Learning ' , Applied sciences (Basel) , vol. 11 , no. 15 , 6912 . https://doi.org/10.3390/app11156912

http://hdl.handle.net/10138/333297
https://doi.org/10.3390/app11156912

applied sciences

MDPI

Article

Identifying Mis-Configured Author Profiles on Google Scholar Using Deep Learning

Jiaxin Tang [1,2], Yang Chen [1,2,*], Guozhen She [1,2], Yang Xu [1], Kewei Sha [3], Xin Wang [1,2], Yi Wang [4,5], Zhenhua Zhang [6] and Pan Hui [7,8]

1 School of Computer Science, Fudan University, Shanghai 200433, China; jxtang18@fudan.edu.cn (J.T.); hazelnutsgz@gmail.com (G.S.); xuy@fudan.edu.cn (Y.X.); xinw@fudan.edu.cn (X.W.)
2 Shanghai Key Lab of Intelligent Information Processing, Fudan University, Shanghai 200433, China
3 Department of Computing Sciences, University of Houston-Clear Lake, Houston, TX 77058, USA; sha@uhcl.edu
4 Peng Cheng Laboratory, Shenzhen 518055, China; wy@ieee.org
5 Institute of Future Networks, Southern University of Science and Technology, Shenzhen 518055, China
6 Meituan, Beijing 100102, China; zhangzhenhua02@meituan.com
7 Department of Computer Science, University of Helsinki, 00014 Helsinki, Finland; panhui@cse.ust.hk
8 Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong
* Correspondence: chenyang@fudan.edu.cn

Abstract: Google Scholar has been a widely used platform for academic performance evaluation and citation analysis. The issue about the mis-configuration of author profiles may seriously damage the reliability of the data, and thus affect the accuracy of analysis. Therefore, it is important to detect the mis-configured author profiles. Dealing with this issue is challenging because the scale of the dataset is large and manual annotation is time-consuming and relatively subjective. In this paper, we first collect a dataset of Google Scholar's author profiles in the field of computer science and compare the mis-configured author profiles with the reliable ones. Then, we propose an integrated model that utilizes machine learning and node embedding to automatically detect mis-configured author profiles. Additionally, we conduct two application case studies based on the data of Google Scholar, i.e., outstanding scholar searching and university ranking, to demonstrate how the improved dataset after filtering out the mis-configured author profiles will change the results. The two case studies validate the importance and meaningfulness of the detection of mis-configured author profiles.

Keywords: Google Scholar; author profiles; mis-configuration; machine learning; neural network; node embedding

Citation: Tang, J.; Chen, Y.; She, G.; Xu, Y.; Sha, K.; Wang, X.; Wang, Y.; Zhang, Z.; Hui, P. Identifying Mis-Configured Author Profiles on Google Scholar Using Deep Learning. *Appl. Sci.* **2021**, *11*, 6912. https://doi.org/10.3390/app11156912

Received: 5 June 2021
Accepted: 20 July 2021
Published: 27 July 2021

1. Introduction

Studies about academic impacts have recently received widespread attention. Web-based publicly available data is widely used for this kind of study [1–3]. After its launch in November 2004, Google Scholar has become one of the largest academic web search engines. It indexes most of the journal and conference papers, theses and dissertations, academic books, technical reports, and other scholarly literature from all broad areas of research. Although there is no official statement regarding the size of the Google Scholar database, it is estimated that the number of articles was about 160 million in May 2014 and the number reached 389 million in January 2018, which is much larger than other academic search engines and bibliographic datasets [4]. Because of its abundant amount of information, most researchers have adopted Google Scholar to fulfill their academic-related search needs. It is widely acknowledged by both academia and industry for paper search and scholar evaluation. A lot of studies about academic impacts are based on Google Scholar data, including co-author analysis [5] and citation analysis [6,7]. Moreover, the metrics such as h-index [8] and total number of citations adopted by Google Scholar are

widely used to evaluate the performance of scholars and institutions [9]. A large number of scholars have created their own author profiles on Google Scholar. A scholar's author profile contains a brief introduction, interested research fields, a paper list, a co-author list and some automatically calculated citation metrics. The author profile is an important reference when evaluating a scholar. It should be noted that the papers added to a scholar's author profile are just claimed by the scholar based on the recommendations given by Google Scholar, without further verification. Moreover, after the author profile is initially established, Google Scholar will continuously add new papers that are believed to be authored by the scholar based on Google Scholar's recommendation algorithm if the author allows. Therefore, some author profiles may turn to be mis-configured even if it used to be reliable.

To make the results of the researches about academic impacts as accurate as possible, some corrections and normalization were applied to raw data from Google Scholar. Most of the previous studies about the reliability of academic data are concentrated on author name disambiguation [10–13]. However, there is little research focusing on the reliability of a scholar's author profile. The mis-configured author profiles may lead to the statistical error of scholars' evaluation metrics and their academic social networks, which may further cause some studies about academic impacts to be inaccurate. The validity of researches mentioned above can be further improved once the author profiles from Google Scholar become more reliable. Therefore, the vulnerability of the information of author profiles is noteworthy, and to detect the mis-configured author profiles, which include papers written by other scholars, is necessary.

In this paper, we focus on the configuration of the author profiles from Google Scholar. We define an author profile that contains papers not authored by the profile owner as a mis-configured one; otherwise it is regarded as a reliable one. We aim at checking the configuration of an author profile automatically and we analyze the data reliability of Google Scholar author profiles according to the judgment results using our proposed model. To the best of our knowledge, it is the first time that the reliability checking has been used upon an academic author profile dataset. Overall, the contributions of our work are summarized as follows:

(1) We reveal the fact that several scholars' author profiles on Google Scholar include papers that are not authored by the scholar, and formulate the problem of detecting mis-configured author profiles on Google Scholar.

(2) We propose and implement a novel neural network model to distinguish mis-configured author profiles on Google Scholar from the reliable ones, with the assistance of graph embedding, classic machine learning, and deep learning techniques.

(3) We construct a dataset of author profiles from Google Scholar and perform a comprehensive data-driven evaluation by applying the model to the real-world dataset. F1-score of the best results reaches 0.9610, which demonstrates the effectiveness of the model.

(4) We conduct an analysis about the mis-configuration phenomenon of Google Scholar by introducing two application case studies and comparing their results using the data before and after filtering out the mis-configured author profiles, which demonstrates the meaningfulness of our work.

The paper is organized as follows. Section 2 defines the problem we will solve. Then, in Section 3 we introduce our dataset and the method to obtain the ground truth to verify the reliability of profiles. In Section 4, we compare the mis-configured author profiles with the reliable ones using some selected features. We introduce our integrated model to identify mis-configured author profiles in Section 5 and evaluate its performance in Section 6. Then, we perform two application case studies and conduct some analysis based on the results of detection in Section 7. In Section 8, we discuss some limitations of our work. Section 9 lists some important related work. Finally in Section 10, we make a conclusion and talk about the future work.

2. Problem Definition

In this section, we first introduce the vulnerability of author profiles. Then, we formally define the problem of author profile mis-configuration.

2.1. Vulnerability of Author Profiles

The paper list of each scholar's author profile on Google Scholar is configured and maintained by the scholar. For convenience, Google Scholar may recommend some papers to a scholar that are likely to be written by her according to her name abbreviation (last name and the initial of the first name). The recommendation is coarse-grained and lack of accuracy. There is a great possibility for a scholar named Alice Wang receiving a paper recommendation written by another author, for example, Angela Wang, since they share the same name abbreviation. If a scholar approves all the recommended papers without proofreading, some papers that were not written by her might be wrongly added to her list, and the paper list will not be further verified by Google Scholar, even if the scholar claims some papers not written by her consciously or unconsciously. Moreover, Google Scholar can continuously add papers to the paper list automatically if the author allows. Many scholars may not actively manage their author profiles on a regular basis. The inaccuracy of scholars' paper lists will seriously diminish the reliability of Google Scholar's author profiles.

2.2. Author Profile Mis-Configuration

Let P^a denote the author profile of scholar a, containing the introduction, interested research fields, a claimed co-author list and a claimed paper list. Some evaluation metrics such as h-index and total number of citations are also a part of the author profile. The claimed co-author list only contains co-authors the scholar wants to display. Denoting the paper list of scholar a as l^a, which consists of M^a paper entries; each paper entry l_i^a in the paper list l^a has its title and an author list with name abbreviations of its authors. Some of the paper entries also have the information of their publication venues, publication years and the numbers of citations. We regard P^a as a mis-configured author profile if there exists at least one paper entry in l^a that does not belong to scholar a. Otherwise, the author profile is treated as a reliable one.

In this paper, given the author profiles of a set of scholars A, we try to judge whether each of them to be mis-configured. We formulate the problem of mis-configuration detection as a binary classification problem.

$$\forall a \in A, f(P^a) \rightarrow \{0, 1\} \tag{1}$$

where $f(P^a) = 1$ means P^a is reliable, while $f(P^a) = 0$ represents that P^a is a mis-configured author profile. An undirected co-author graph $G = (V, E)$ can be constructed based on the paper lists of scholars A to assist the mis-configuration detection, where each node $v \in V$ denotes a scholar and each edge $e \in E$ represents the collaboration between two scholars. Due to the large scale of the dataset, the detection system must be automatic and scalable. Note that, in this paper, we do not attempt to explicitly find the papers which are wrongly claimed in the paper list and we leave this for the future work.

3. Data Collection

In this section, we describe the methods to obtain the data from Google Scholar and provide an overview of the dataset.

3.1. Author Profiles on Google Scholar

In this paper, we focus on the mis-configuration detection of author profiles on Google Scholar. Since there is no existing public dataset incorporating the author profiles on Google Scholar, we try to collect the author profiles of scholars in the field of computer science by

ourselves. In order to improve efficiency, we design a distributed crawler, which contains a Redis server on the master node and several slave nodes with Redis clients.

The search engine of Google Scholar allows searching for scholars according to their research fields. We resort to the "Metrics" page provided by Google Scholar, which gives a ranking of the publications. Under the Engineering & Computer Science domain, there are 58 sub-domains. Filtering out the irrelevant sub-domains, 170,483 scholars' profiles from 31 sub-domains are collected in the end. We establish a comprehensive dataset of scholars in the field of computer science. Therefore, we can build a co-author graph to facilitate the detection of mis-configuration. Moreover, we will analyze the impact of mis-configuration in the field of computer science from a macro perspective in Section 7. Notice that our dataset only includes publicly available data from Google Scholar.

3.2. Ground Truth

There is no existing mis-configuration label of the scholars' author profiles. Due to the strict requirements of accuracy and veracity, we do not resort to any automatic tool, but leverage a manual labeling system to obtain the ground truth. Although our dataset is limited to the author profile pages on Google Scholar, we resort to some extra information during the manual labeling, such as the papers' abstracts and the homepages of the authors. We randomly select 3000 scholars in the dataset and recruit three students majoring in computer science to label each scholar's author profile. In order to guarantee the reliability of the labeling result, each scholar is labeled by at least two students. If two students give conflicting labels to an ambiguous instance, the third student will check the scholar's author profile and decide the result finally. To make the evaluation criteria of labeling as consistent as possible, we establish a workflow for the manual annotation as follows and train the participated students to follow it:

- Check whether the papers' titles and the publication venues are related to the scholar's claimed research labels. If not, and there is no other papers in the list have the same co-authors as this paper, the paper may be wrongly claimed.
- Pay attention to the scholar's full name and the name abbreviation of each paper in the paper list. Since Google Scholar provides paper recommendations according to the scholar's last name and the initial of her first name, there is a high possibility that a scholar with a common name has a mis-configured author profile.
- Click into the detailed page of each paper in the paper list, which contains the authors, abstract, publication information and the citations. Check whether the scholar is included in the author list. However, scholars may change their full name, especially for female scholars who may change their surname after marriage. Therefore, if the scholar's full name does not exist in the author list, the author profile might still be reliable.
- Check the publication year of each paper. If the time interval between two papers is too long (e.g., the time span more than a century), the possibility of the author profile being reliable is very small. In our labeled dataset, there are 33 author profiles with two papers that are published over 100 years. However, considering there may be some intrinsic mistakes in the database of Google Scholar, some of them might still be labeled as reliable.
- Check the homepage of the scholar if she puts it on the author profile and compare the publications on her homepage with the claimed paper list.

According to the results of the labeling, among the selected 3000 scholars' profiles, 90.5% (2715) of them are reliable and 9.5% (285) of them are mis-configured. Limited by the heavy workload of labeling, we get the ground truth of a representative subset of the dataset. The complete dataset provides more comprehensive features, which can further improve the performance of our detection.

4. Comparison between Reliable Author Profiles and Mis-Configured Author Profiles

We compare the reliable author profiles with the mis-configured ones in this section to show their differences. We extract some features from the author profile, and then show the differences between these two kinds of profiles on these features.

4.1. Features of an Author Profile

In this subsection, we elaborate some features manually extracted from the author profile including features of basic information, features of evaluation metrics and features of research topics.

4.1.1. Features of Basic Information

- #papers: The total number of papers in the paper list represents the scholar's productivity. However, scholars who claimed a large number of papers are more likely to include mis-configured papers unconsciously if they are not senior enough.
- #Research fields: A list of interested research fields is displayed on the author profile. We count the number of interested research fields that the scholar claimed. Limited by the time and energy, there is little possibility for a decent scholar to focus on too many fields unless she works in interdisciplinary areas.
- Verified: A scholar can verify the author profile through an institutional email address. Whether the account is verified is also an important feature. Scholars who are verified by institutional email addresses will earn more reliability from the reputation of the institution.
- #co-authors: We count the number of co-authors with different names in the paper list. This feature might not reflect the exact number of co-authors, because different scholars may have the same name abbreviation. However, it can still provide an approximate size of the scholar's collaborators.
- Time span: Most of the papers claimed in the paper list have publication years. We calculate the time span of author a ranging from the first paper published to the latest paper as follows:

$$timeSpan = \max_{i,j \in M^a} (|y_i^a - y_j^a|) \tag{2}$$

where y_i^a denotes the publication year of paper i authored by author a. Due to the limited life span of the scholar, the author profile may be mis-configured if the value of this feature is too large. However, publication year of each paper in the paper list is automatically generated by Google Scholar according to its database, there may be some mistakes. So, it is unreasonable to set a clear threshold for this feature.

4.1.2. Features of Evaluation Metrics

- H-index, i10-index [14]: H-index and i10-index are indicators of the number of high-quality papers, which are important to quantify the actual productivity and apparent scientific impact of a scholar. Scholars with higher h-index are considered to have greater impact. They tend to pay more attention to their author profiles. However, h-index can also measure the productivity of the scholar. Higher h-index means more papers, which may lead to a higher possibility of mis-configuration.
- #citations: The feature is the total number of citations earned by all the papers in the paper list. The impact of this feature is similar to h-index.
- citation-std: The citation of each paper can reflect its quality to some extent. So, we calculate the standard deviation of the numbers of citations of all papers. Intuitively, the quality of the papers written by a scholar may be similar.

4.1.3. Features of Research Topics

For a certain scholar, their research topics can be extracted from the title and publication venue (for argument's sake, we call the concatenation of paper title and publication

venue as paper entry in this subsection) of each paper in the paper list. We first utilize Google Cloud Translation API [15] to translate all the paper entries into English. Then, we leverage Latent Dirichlet Allocation (LDA) [16], a generative statistical model that can learn the topic-specific word distribution, to get the probability distribution of each paper entry on different topics. LDA allocates each word in the paper entries with a latent variable choosing an underlying topic. Latent topic variables t has a probability distribution $\theta_d = p(t|d)$ over paper entry d, and word w is associated with a distribution $\phi_t = p(w|t)$ over topic variable t. Finally, we obtain the probability distribution of each paper entry on 30 topics.

Limited by the time and interest, a scholar can only focus on a few related topics. We calculate some metrics representing the diversity of a scholar's paper list, which indicates the broadness of the scholar's research fields. We choose the symmetric cross-entropy divergence to represent the difference between two paper entries. For example, the difference between paper entry i and paper entry j is calculated as follows, where t means the topics obtained from LDA and $p(t_i)$ represents the probability of paper entry i to be related to topic t:

$$symCrossEntropy_{i,j} = \sum_t p(t_i) log \frac{1}{p(t_j)} + \sum_t p(t_j) log \frac{1}{p(t_i)} \quad (3)$$

We calculate the maximum, average, and standard deviation values of symmetric cross-entropy between each two paper entries of a scholar to characterize the diversity.

$$maxCrossEntropy_a = \max_{i,j \in M^a} symCrossEntropy_{i,j} \quad (4)$$

$$meanCrossEntropy_a = \prod_{i,j \in M^a} (symCrossEntropy_{i,j}) \quad (5)$$

$$crossEntropyStd_a = \sigma_{i,j \in M^a} (symCrossEntropy_{i,j}) \quad (6)$$

Basically, papers of a single scholar can be divided into several clusters corresponding to different research projects. Therefore, we use non-negative matrix factorization (NMF) [17] to cluster the paper entries in the paper list according to their probability distribution on 30 topics and use silhouette score [18] to measure the distance between different clusters. We use the maximum silhouette score as the feature.

Finally, the features we obtain from the scholars' author profiles are listed in Table 1, in which the meaning of the last two columns will be explained in detail in the next subsection. p-value is calculated by a two-sided t-test on two kinds of author profiles, and information gain (IG) is computed for feature selection.

Table 1. Feature list.

Feature	Description	p-Value (t-Test)	Information Gain
#papers	Number of papers in the paper list	3.51×10^{-12}	0.039
H-index	H-index of the scholar	1.20×10^{-27}	0.047
i10-index	i10-index of the scholar	6.26×10^{-16}	0.047
#citations	Total number of citations for all papers	7.89×10^{-20}	0.036
citation-std	Standard deviation of the number of citations	8.32×10^{-6}	0.023
#Research fields	Number of interested research fields	0.35	0.083
Verified	Whether verified using institutional email address	/	/
Time span	Maximum time interval between two papers	1.73×10^{-4}	0.060
#co-author	Number of co-authors with different names	7.40×10^{-11}	0.044
maxCrossEntropy	Maximum cross-entropy between two paper entries	5.77×10^{-17}	0.024
meanCrossEntropy	Average cross-entropy between two paper entries	2.92×10^{-3}	0.020
crossEntropyStd	Standard deviation of the cross-entropy between two paper entries	5.04×10^{-6}	0.021
silhouette score	The maximum silhouette score of the NMF clustering based on research topics	8.13×10^{-32}	0.041

4.2. The Differences between Reliable and Mis-Configured Author Profiles

We analyze features of reliable author profiles and mis-configured ones to demonstrate their differences. It is noteworthy that we only discuss the differences in the overall distribution without talking about each individual. For each feature, we show the distributions of reliable author profiles and mis-configured ones (Figure 1).

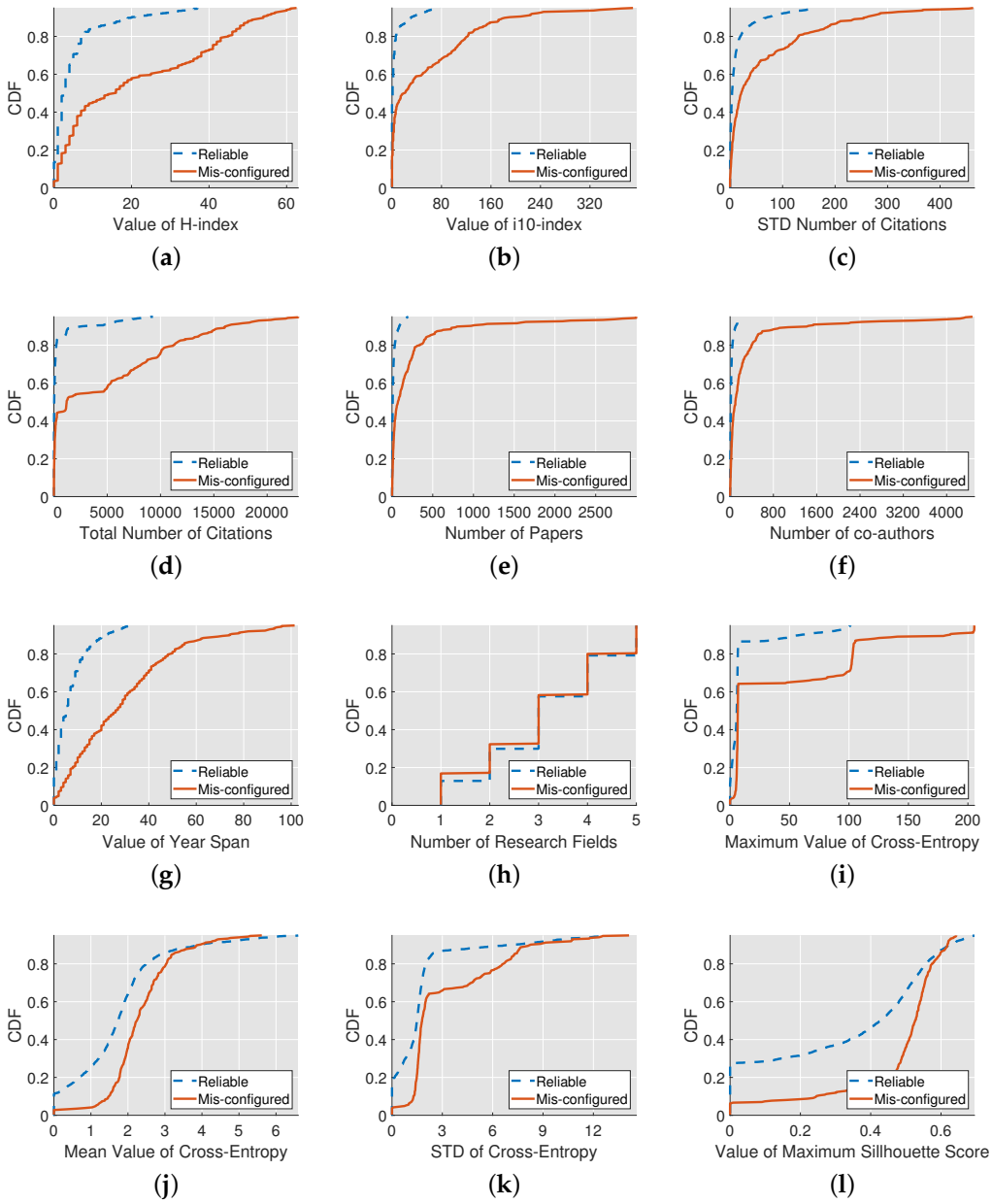


Figure 1. Comparison between reliable author profiles and mis-configured author profiles on different features. (**a**–**l**) show the comparison on features in Table 1 (except Verified, which is a two-value feature), respectively.

We can find that the mis-configured author profiles mostly have much larger numbers of papers in their paper lists compared with the reliable ones. Furthermore, the total numbers of citations are also larger. For half of the reliable author profiles, the standard deviation of citation numbers are smaller than 4, while the value is larger than 21 for mis-configured author profiles. From the cross-entropy between topic distributions of different paper entries and the value of maximum silhouette score, we find that the research topics on

the mis-configured author profiles are relatively richer and more diverse. Furthermore, the largest time intervals between two papers are longer. Moreover, the h-index and i10-index of the mis-configured author profiles are relatively higher. The h-index values of 83% of the reliable author profiles are lower than 10, while only 44% of the mis-configured author profiles have h-index less than 10. The median value of other i10-index metric is only 1 for the reliable author profiles. However, the median value for the mis-configured ones is 21. Since these two metrics are widely used to evaluate a scholar, this phenomenon can partly explain the possible purpose for some intentional mis-configuration.

We conduct a two-sided *t*-test [19] on these two kinds of author profiles. The *p*-value is shown in the third column of Table 1. The values of all these features except the number of claimed interested research fields are less than 0.05, which means that the difference between two kinds of author profiles is significant on these features. As for the research fields, most of the scholars list less than five tags, so this feature has relatively smaller differences between reliable and mis-configured profiles. Scholars with reliable profiles may be more rigorous to maintain it. Therefore, the number of listed research fields may be more focus on three to five. However, more mis-configured author profiles have less than three research fields.

Owing to these differences, these features can in turn assist the mis-configuration detection. To evaluate their effects quantitatively, we calculate the information gain (IG) [20] for further demonstration. The result is shown in the last column of Table 1. The larger the IG value is, the more important the feature is for classification. According to IG values, we find that the standard deviation of citations (i.e., citation-std) and the features of research topics between two single paper entries (i.e., maxCrossEntropy, meanCrossEntropy, crossEntropyStd) would make relatively less contributions to the classification. So we use the other nine features to assist the detection of mis-configured author profiles.

5. Mis-Configuration Detection Model

Mis-configuration of author profiles can seriously damage the reliability of Google Scholar, and then limit further applications of the data from it. Since manually labeling all the scholars' author profiles is time-consuming and relatively subjective, we propose a model to separate mis-configured author profiles from reliable ones automatically.

5.1. Model Overview

We propose a unified model which integrates the deep learning technologies with classic machine learning classification algorithms to identify the mis-configured author profiles. The model architecture is shown in Figure 2.

In a nutshell, the model has four modules. First, we leverage TextCNN [21], which is known to be good at extracting local features from texts, for paper title encoding as it is shown in the green box (Section 5.2). Then the latent features of all the paper titles in a paper list are sorted into a sequence. The sequential features are captured using a RNN-based method, CW-RNN [22], and are aggregated by a paper attention-based aggregator. The process is shown in the purple box (Section 5.3). Moreover, in the orange box, the social feature of each scholar is extracted from the co-author graph (Section 5.4). The social graph embedding is obtained using a state-of-the-art method for node embedding, i.e., node2vec [23]. Finally, the aggregated feature of the paper list, the social graph embedding and the selected features mentioned in Section 4.1 are fed into a supervised machine learning-based classifier to find whether the author profile is mis-configured or reliable (Section 5.5). The pseudocode of the model training and test process is shown as Algorithm 1. We will introduce the four modules, respectively, in detail in the following subsections.

Algorithm 1 The pseudocode of the mis-configuration detection model.

Input: paper lists of all the author profiles: X_{paper_list} (the paper list of each author profile consists of several paper entries), co-author graph: G, selected features of all the author profiles: $X_{selected\_features}$, the maximum number of papers in the paper list: n_a

Output: Mis-configuration classification result of each user: *res*

Initialize parameters
for each training iteration **do**
  Sample a batch of training data
  for $X_{paper\_list}[i] \in X_{paper\_list}[batch]$ **do**
    for $paper\_entry[j] \in X_{paper\_list}[i]$ **do**
      $paper\_embedding[i][j] \leftarrow TextCNN(paper\_entry[j].title)$
    end for
  end for
  $paper\_list\_feature[batch] \leftarrow$ CW-RNN$(paper_embedding[batch][1 \dots n_a])$
  $social\_embedding[batch] \leftarrow Node2vec(G)$
  $feature[batch] \leftarrow paper\_list\_feature[batch] \bigoplus social\_embedding[batch]$
  $feature[batch] \leftarrow feature[batch] \bigoplus X_{selected\_features}[batch]$
  $nn\_res[batch] \leftarrow Fully\_Connected\_NN(feature[batch])$
  Update the neural network parameters
  $res[batch] \leftarrow LightGBM(feature[batch])$
  Update the LightGBM parameters
end for
for each test iteration **do**
  Sample a batch of testing data
  for $X_{paper\_list}[i] \in X_{paper\_list}[batch]$ **do**
    for $paper\_entry[j] \in X_{paper\_list}[i]$ **do**
      $paper\_embedding[i][j] \leftarrow TextCNN(paper\_entry[j].title)$
    end for
  end for
  $paper\_list\_feature[batch] \leftarrow$ CW-RNN$(paper_embedding[batch][1 \dots n_a])$
  $social\_embedding[batch] \leftarrow Node2vec(G)$
  $feature[batch] \leftarrow paper\_list\_feature[batch] \bigoplus social\_embedding[batch]$
  $feature[batch] \leftarrow feature[batch] \bigoplus X_{selected\_features}[batch]$
  $res[batch] \leftarrow LightGBM(feature[batch])$
end for

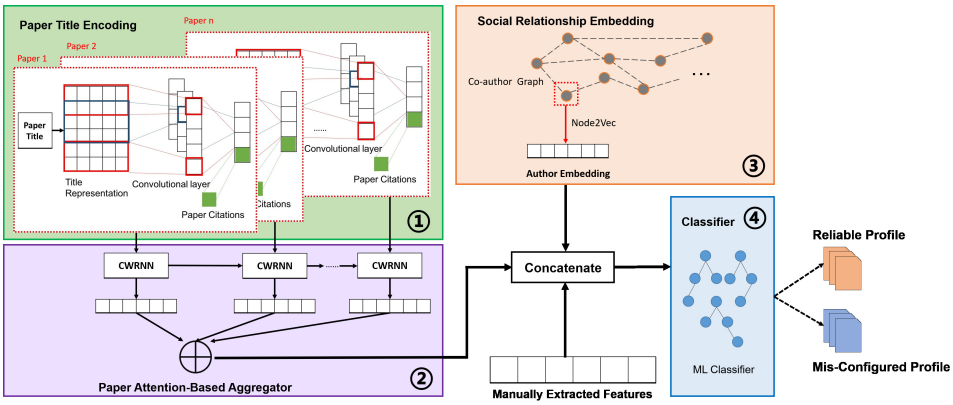


Figure 2. The architecture of the model. (1) The green box (box ①) is the process of paper title encoding where TextCNN is leveraged. (2) The paper attention-based aggregator is shown in the purple box (box ②), where the hidden features of different papers are produced by CW-RNN and aggregated with different weights. (3) The orange box (box ③) shows the process to obtain the author embedding from the co-author graph. (4) The blue box (box ④) is a machine learning classifier (we use LightGBM in this paper) to classify the author profiles based on the concatenated features. (5) The output of the models is the judgment of whether an author profile is mis-configured or not.

5.2. Paper Title Encoding

According to our definition of mis-configured author profiles, paper information in the paper list is the most intrinsic feature for mis-configuration detection. Therefore, we first try to deal with the titles of papers in the paper list. Since hand-crafted feature engineering on textual features is labor-intensive and subjective, we regard the title of each paper as a sentence and use a deep learning method to automatically extract features from the raw text. We leverage TextCNN [21], which is first proposed by Kim. The model architecture consists of an embedding layer, a convolution layer and a max-pooling layer. The embedding layer is designed to transfer the one-hot representation of each word to a low-dimensional vector. A title with n words is represented as follows:

$$\boldsymbol{w}_{1\ldots n} = \boldsymbol{w}_1 \oplus \boldsymbol{w}_2 \oplus \ldots \oplus \boldsymbol{w}_n \tag{7}$$

where $\oplus$ is the concatenation operation and $\boldsymbol{w}_i$ represents the embedding vector of word i. To deal with titles with different lengths, we introduce zero-paddings to paper titles. Then, a convolution layer gets local features using multiple filters with different sizes and a max-pooling layer for reducing dimension. For the ith filter with size h, the local feature e_i is calculated using the following equations:

$$\boldsymbol{c}_{i,j} = ReLU(\boldsymbol{f} \cdot \boldsymbol{w}_{j\ldots j+h-1} + \boldsymbol{b}) \tag{8}$$

$$\boldsymbol{e}_i = \max\left(\boldsymbol{c}_{i,1}, \boldsymbol{c}_{i,2}, \ldots, \boldsymbol{c}_{i,n-h+1}\right) \tag{9}$$

Finally, the outputs of the max-pooling layer e of all the filters are seen as the latent feature of each paper title. Compared with LDA-based methods, which only consider the topic of each single word, TextCNN can capture the context feature in the sentence.

5.3. Paper Attention-Based Aggregator

We concatenate the latent feature of paper title with the number of citations. The condensed vector represents the feature of a single paper. To extract the sequential feature of the paper list, we first sort the papers of each scholar in her paper list by their publication time. For those papers do not have a publication time, we regard them as the earliest ones. Then, RNN-based methods are applied to the sequential data. Here, we choose CW-RNN [22] to get the hidden feature of each paper. It solves long-term dependency problem by dividing the RNN hidden layer into different modules, which run at different clock speeds. At each step t, only the modules whose clock speeds can be divisible by t are executed. This method reduces the number of RNN parameters and makes the training process faster. Please note that not all the papers in the paper list contribute equally to the detection of mis-configuration. Therefore, we introduce an attention mechanism to aggregate the hidden features of all papers. The final latent feature of a paper list claimed by scholar i is calculated as follows:

$$\boldsymbol{s}^{\boldsymbol{i}} = \sum_{t=1}^{M^i} \alpha_t^i \boldsymbol{h}_t^i \tag{10}$$

where M^i is the number of papers in scholar i's paper list. $\boldsymbol{h}_t^i$ denotes the hidden feature of the tth paper in scholar i's paper list. α_t^i measures the importance of the tth paper, and it is computed as follows:

$$\boldsymbol{u}_t^i = tanh(\boldsymbol{W}_w \boldsymbol{h}_t^i) \tag{11}$$

$$\alpha_t^i = \frac{exp(\boldsymbol{u}_t^i \boldsymbol{u}_w^T)}{\sum_{k=1}^{M^i} exp(\boldsymbol{u}_k^i \boldsymbol{u}_w^T)} \tag{12}$$

where $\boldsymbol{W}_w$ is a parameter. $\boldsymbol{u}_t^i$ is the hidden representation of $\boldsymbol{h}_t$ and $\boldsymbol{u}_w$ is the weight parameter to represent the paper-level context vector.

5.4. Social Relationship Embedding

Apart from the features of paper title and citations, we also leverage the co-author relationship to get the social feature of each scholar considering its sustainability [24]. The co-author information reflects social competence of different authors, which also indicates the reputation. The credit of the collaborators can enhance the reliability of the scholar. In this part, we try to obtain a social graph embedding of the scholars based on the co-author relationship.

In order to characterize the social feature of each scholar, we build a co-author graph $G = (V, E)$ with a node $v \in V$ denoting a scholar and an edge $e \in E$ representing the collaboration between two scholars. The weight of an edge represents the number of papers co-authored by the two collaborators.

The co-author graph is constructed based on the paper lists of all the scholars in our dataset. We scan all the scholars and count the authors of the distinct 6,202,775 articles. In order to deal with duplication of paper's title, we use the tuple (title, authors and publication information) to identify a paper and the hash technique is used to accelerate the counting process. Then, we add an edge between every two authors of a paper. Finally, we obtain a graph with 170,483 nodes and 1,756,688 edges. We utilize node2vec to get the embedding representation of each node, which is defined as the social features of a scholar. Node2vec is a framework to learn the feature representation for nodes in the graph. It is leveraged because of its remarkable ability to deal with large-scale graphs and preserve the network neighbors of nodes. It is worth mentioning that other node embedding algorithms can also be used here.

5.5. Profile Classifier

Finally, we try to take advantage of the tree-structure classifiers to build an integrated model which combines the neural network model with the classic machine learning algorithms. Since the classifier used in the neural network just contains multiple linear layers with a nonlinear activation function, its classification performance greatly depends on the number of neurons, while a large number of neurons require a lot of computing resources. Moreover, it is challenging to determine hyperparameters that can generate the best classifier on training set without overfitting. We utilize a LightGBM classifier [25] for its high and stable performance. LightGBM is a gradient boosting framework, which is much more efficient than most existing implementations of gradient boosting trees.

The input of the classifier is the concentration of the final latent feature of the paper list, social graph embedding of the scholar, and the features we extracted and selected in Section 4.1. We normalize these features to $[0, 1]$ by the following formula, where x_i denotes the feature x of the scholar i and $max(x)$, $min(x)$ denote the maximum and minimum elements of feature x, respectively.

$$x_i^{normalized} = \frac{x_i - min(x)}{max(x) - min(x)} \tag{13}$$

The output of our proposed neural network model can be seen as the probability for the scholar's author profile to be reliable or mis-configured.

We design a two-step process to train the classifier. First, we use a two-layer fully-connected neural network classifier, which is hardly tuned to train the parameters of paper title encoding module and paper attention-based aggregator module. It is noted that the numbers of papers from different scholars vary considerably. In our labeled dataset, the largest number of papers of one scholar is 2960, while the smallest number is only one. Mini-batch [26] technique is utilized due to the limitation of memory and can help to accelerate the training process. Dropout [27] is used to avoid overfitting.

According to the result of our data labeling in Section 3.2, the dataset is imbalanced, which means that the number of reliable author profiles is larger than the number of mis-configured ones. Since an author profile may be mis-configured due to only a few papers in the paper lists, the feature spaces of reliable and mis-configured profiles may

have overlapping regions and be easy to confuse. So, instead of using resampling methods to adjust the data proportion of the original dataset, which may introduce noise into the training process, we try to improve the loss function of the algorithm to deal with the problem of imbalanced data. The designed loss function to balance the importance of positive and negative samples are shown as follows:

$$loss = \frac{1}{N_R} \sum_{a \in R} \sum_{i \in \{0,1\}} I(i) log(p(i)) + \frac{1}{N_M} \sum_{a \in M} \sum_{i \in \{0,1\}} I(i) log(p(i)) \quad (14)$$

where R, M denote the set of reliable and mis-configured author profiles, respectively. Then, N_R, N_M mean the numbers of reliable and mis-configured author profiles. $p(0)$ means the probability to be mis-configured, and $p(1)$ is the probability to be reliable, which are the output of the network. $I(\cdot)$ is the indicator function, and its value can be set as Equation (15). Finally, we use the concatenated features trained from the neural network model to further train the LightGBM classifier.

$$I(i) = \begin{cases} 1 & the\ author\ profile\ \text{i}\ is\ reliable \\ 0 & the\ author\ profile\ \text{i}\ is\ mis\text{-}configured \end{cases} \quad (15)$$

6. Results

To evaluate the performance of our proposed model, we conduct extensive analysis. First, we compare it with some baseline methods. Then, we attempt to analyze the benefit of each component of the model and the contributions of different features.

6.1. Experimental Settings

We split the labeled dataset into a training set and a test set, and the fraction of them is 2:1. We use F1-score as the metric to evaluate the performance of each method. F1-score is the harmonic mean of precision and recall. Precision represents the fraction of predicted reliable author profiles are indeed reliable, while recall means the fraction of reliable author profiles which are predicted correctly. We use 10-fold cross-validation to determine the most suitable hyperparameters of LightGBM. The hyperparameters of neural network part are set as follows: $embedding\_dim = 200$, $num\_filters = 150$, $kernel\_size = \{2, 3, 4\}$, $cwrnn\_hidden\_dim = 10$, $dropout = 0.5$. Additionally, we also pay attention to AUC (Area Under Curve) which is the area under the ROC curve [28]. The value of AUC is equal to the probability that the classifier will rank a randomly chosen positive example (reliable author profile) higher than a randomly chosen negative example (mis-configured author profile).

6.2. Performance Comparison

6.2.1. Baselines

We compare the performance of our proposed model with some baselines, which include the state-of-the-art classification algorithms, a sybil detection method (SybilSCAR) and a text classification method (HAN).

- LR [29]: Logistic Regression (LR) is a simple classification model that adds a logistic function upon the linear regression. We feed the selected features in Section 4.1 into LR to predict the labels.
- DT [30]: Decision tree (DT) is one of the classic machine learning algorithms, which divides the final classification outcomes into a series of related choices on the features. Here, we use DT to separate the mis-configured authors profiles and reliable ones based on the selected features mentioned in Section 4.1.
- RF [31]: Random forest (RF) utilizes an ensemble of DTs trained with the bagging method to make a more accurate and stable prediction. Selected features mentioned in Section 4.1 are used for classification.
- XGBoost [31]: XGBoost is a scalable end-to-end tree boosting system widely used in the machine learning competitions. It provides a parallel boosting, which is highly efficient, flexible and portable, to solve the classification problem.

- LightGBM: We leverage LightGBM for mis-configuration detection based on the features extracted and selected in Section 4.1.
- SybilSCAR [32]: SybilSCAR is a state-of-the-art algorithm for scalable sybil detection that unifies Random-Walk-based and Loop-Belief-Propagation-based methods. It is suitable for large-scaled networks and is robust to noise. We apply this method to the co-author network and the prior probability of each node to be mis-configured is obtained by XGBoost.
- HAN [33]: HAN is a hierarchical attention neural network framework to deal with the content data, which is originally proposed for fake news detection. We leverage this method based on the paper list of each scholar.

6.2.2. Performance Comparison

The comparative results are shown in Table 2. We can find that our proposed model achieves the best performance with the assist of the NLP features of paper lists and the co-author-based social features. We use McNemar's test [34] to calculate the statistical significance between our method and the baselines. The result shows that our proposed method is significantly different from each of the baseline methods (p-value < 0.001). By comparing different baselines, it is easy to find that the machine learning models outperform the graph-based sybil detection method (i.e., SybilSCAR) and document classification method (i.e., HAN), which reflects that the features we extract and select are more effective to identify mis-configured author profiles. In particular, the performance of SybilSCAR is not very good, which indicates that the similarity between scholars who have co-author relationship does not play a crucial role in the detection and semi-supervised learning based on the co-author graph cannot detect mis-configuration effectively. The NLP features and the co-author-based social features can also contribute to the detection; however, the improvement is not as great as expected. We believe that there are two reasons that may explain it: Firstly, some mis-configured author profiles only contains a small number of wrongly claimed papers, which makes it difficult to detect. Secondly, interdisciplinary research is a hot spot. Some scholars may collaborate with scholars from different fields to do research on various topics. Therefore, papers in one scholar's list may be diverse, which makes it hard to judge without other references.

Table 2. Performance comparison with baselines.

Methods	Precision	Recall	F1-Score	AUC	p-Value (McNemar's Test)
Linear Regression	0.9220	0.9912	0.9553	0.5913	2.921×10^{-9}
Decision Tree	0.9237	0.9890	0.9552	0.7664	3.283×10^{-8}
Random Forest	0.9313	0.9735	0.9520	0.7570	5.702×10^{-7}
XGBoost	0.9296	0.9768	0.9526	0.8062	2.899×10^{-7}
LightGBM	0.9306	0.9768	0.9532	0.7835	1.439×10^{-7}
SybilSCAR	0.4803	0.9946	0.6477	0.6148	1.623×10^{-28}
HAN	0.9227	0.9095	0.9161	0.6301	2.896×10^{-4}
Our Proposed Model	0.9307	0.9934	0.9610	0.8200	/

In conclusion, our model synthesizes the NLP feature of each paper, co-author-based social relationship and the selected features on the author profile page to justify the reliability of the author profile, which is more comprehensive and accurate compared with the baseline methods. The paper title encoding module can effectively characterize the NLP feature of the title content. The paper attention-based aggregator can integrate the features of each paper and capture the sequential feature of the whole paper list. Node2vec can obtain the social feature based on the co-author relationship. Finally, the tree-structure classifier is effective to predict the reliability of the author profile based on all the features. To further demonstrate the contributions of these modules, we conduct more experiments in the next subsection.

6.3. Component-Wise Evaluation

To evaluate and quantify the contribution of each component of our proposed model, we compare them with some other alternative modules.

Paper Attention-Based Aggregator: First, we focus on the paper attention-based aggregator module. We compare the "CW-RNN w/Attention" with sole "CW-RNN" and sole "Attention". Then we replace the CW-RNN model with two other RNN-based methods which are commonly used, i.e., LSTM [35] and Dilated RNN [36], to obtain the sequential features. LSTM uses a specialized architecture which contains an input gate, an output gate and a forget gate to mitigate the problem of vanishing and exploding gradients. Thus, it handles both long-term and short-term characteristics. Dilated RNN utilizes a multi-layer and cell-independent architecture characterized by multi-resolution dilated recurrent skip connections to learn sequential dependencies of different scales at different layers. The results are shown in Table 3. We only compare their performance with a fully-connected neural network classifier here.

Table 3. Performance of different aggregators.

Methods	Precision	Recall	F1-Score
CW-RNN	0.9689	0.7544	0.8482
Attention	0.9711	0.6667	0.7906
LSTM w/Attention	0.9689	0.7572	0.8501
Dilated RNN w/Attention	0.9612	0.7357	0.8334
CW-RNN w/Attention	0.9508	0.8513	0.8983

With the assistance of model ablation, we validate the contributions of sequential features and attention mechanism. Attention mechanism can effectively select the more critical features in detecting the mis-configured author profiles. Additionally, we can find that CW-RNN outperforms LSTM and Dilated RNN in capturing the sequential features of the paper list, which indicates that a scholar may focus on several projects at the same time and the sequential features can be extracted from the papers with some intervals after sorting.

However, our aggregator still has some deficiencies. The publication time in the list is only accurate to year which is coarse-grained. Moreover, the publication information of some papers are incomplete. These issues may account for the inaccuracy of the paper sorting. The third-party information might be able to make up these deficiencies and we leave it for the future work.

Classifier: Then we also compare different classifiers to show the effectiveness of LightGBM classifier from the experimental perspective. The algorithms we compare here can be divided into single learning models including LR and DT, and ensemble learning models such as RF and XGBoost. The results are shown in Table 4. All the non-neural network classifiers outperform the fully-connected classifier. Our proposed model introduces the advantages of these classic models into the neural network model and build an integrated model. LightGBM achieves the best performance.

Table 4. Results of prediction using different classifiers.

Methods	Precision	Recall	F1-Score
FC	0.9508	0.8513	0.8983
LR	0.9261	0.9823	0.9534
DT	0.9292	0.9845	0.9561
RF	0.9216	0.9989	0.9587
XGBoost	0.9330	0.9834	0.9575
LightGBM	0.9307	0.9934	0.9610

Ablation Study: To further evaluate the contributions of social graph embedding and manually selected features, we conduct an ablation study. For the sake of convenience, we only compare their performance with a full-connected neural network classifier. In Table 5,

"Model w/o Social" means the neural network without the social relationship embedding and "Model w/o Manual" means the model without the manually selected features. We can find that the performance will be degraded if either of these components is removed.

Table 5. Performance of the ablation study.

Methods	Precision	Recall	F1-Score
Model w/o Social	0.9657	0.6817	0.7992
Model w/o Manual	0.9438	0.7764	0.8520
Our Proposed Model	0.9508	0.8513	0.8983

7. Application Case Study

In this part, we discuss two application case studies, i.e., outstanding scholar searching and university ranking, based on the judgment results of the author profiles in the dataset. The performance of these two case studies before and after filtering out the mis-configured author profiles are compared, which can show the importance of our work.

7.1. Case Study 1: Outstanding Scholar Searching

Google Scholar provides the service that users can search for scholars according to their research fields. The searching results are sorted by each scholar's total number of citations and listed by 10 scholars per page. Since users tend to view the first several pages, the reliability of author profiles with relatively high numbers of citations is much more important. We treat the scholars on the first 10 pages, namely the top 100 scholars of each research field, as the outstanding scholars. In this part, we pay attention to these outstanding scholars and analyze the mis-configuration of their author profiles.

We choose the 10 research fields with the largest numbers of authors in our dataset for analysis, which is listed in Table 6. The top 100 scholars of these 10 research fields who are listed in the first 10 pages are observed. We count the mis-configured author profiles of these 100 scholars. As shown in Figure 3, each of the 10 fields has more than 15 author profiles to be mis-configured. In particular, there are 3 fields with over half of the author profiles being mis-configured. Since the first searching result page is the most important page and most frequently viewed, we also display the number of mis-configured author profiles on the first page in Figure 3 and the numbers of mis-configured ones are excessively large. We believe that a scholar who has a large number of citations tends to have lots of papers in her paper list, therefore is more likely to include others' papers by mistakes. Interestingly, the number of mis-configured author profiles belonging to scholars that list natural language processing (NLP) or data science in their research fields are smaller than that of other eight fields. Intuitively, it is less common for research in NLP to combine with other fields, so the possibility of including others' papers is small. Researches in some other fields such as data mining are similar to those in data science. The reason for such a small number of mis-configured author profiles is that maybe the term is much more professional and the scholars who tend to include this term in their research field lists are more rigorous.

Due to the critical impact of the outstanding scholars, we conduct a comprehensive analysis on the mis-configuration of these scholars' author profiles. Due to the overlapping of scholars in different fields, there are totally 865 outstanding scholars in the first ten pages of these ten research fields. Of them, 341 have mis-configured author profiles. First, we observe the academic age distribution of these scholars. We can estimate the academic age of a scholar by observing the largest time interval between two papers in her paper list. If the time interval is larger than 40 years, we regard her as a relatively senior scholar. Of these scholars, 46.9 percent are senior. Among the senior scholars' author profiles, 70.4 percent are mis-configured and 83.9 percent of mis-configured author profiles belong to the senior scholars, which indicates that scholars might not actively manage their author profiles after retirement. Then, we study the country distribution of these outstanding scholars. Utilizing the institutional email addresses shown on the author profiles, we can also detect

which country each scholar works in. We extract the email domain from the email address and then we query domain name servers (DNS) to obtain its corresponding IP address. GeoLite2 database [37] is used to translate IP address to country information. The country distribution of these outstanding scholars is shown in Figure 4a. After the mis-configured author profiles are filtered, the distribution is displayed in Figure 4b, which becomes more balanced among different countries. There are also some changes in the ranking of the number of scholars. For example, there are relatively more mis-configured author profiles in China. We believe that there are two main reasons to explain this phenomenon. Firstly, the naming convention in China may lead to more name abbreviation duplication. Secondly, many scholars in China cannot easily access Google. Therefore, many of them may not manage their author profiles regularly.

Table 6. Number of scholars in the top 10 research fields.

Research Field	Number of Scholars
machine learning (ML)	53,850
computer vision (CV)	20,354
data mining (DM)	17,225
image processing (IP)	16,166
robotics (Rb)	12,577
computer science (CS)	9100
natural language processing (NLP)	8235
data science (DS)	7317
big data (BD)	7104
pattern recognition (PR)	6095

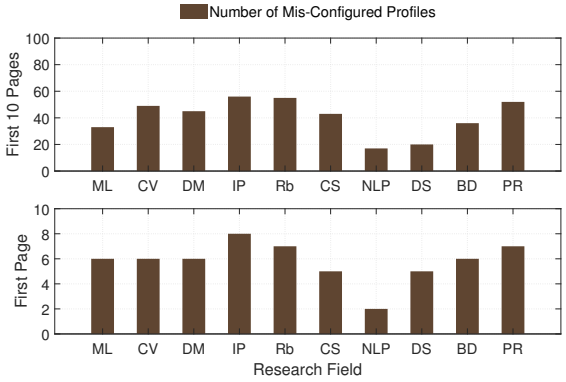
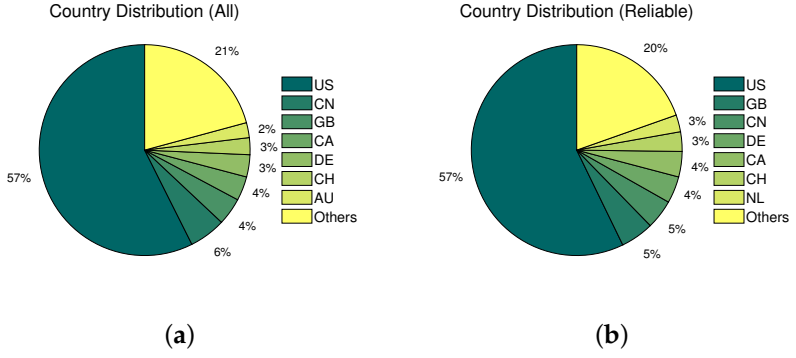


Figure 3. Number of outstanding scholars' mis-configured author profiles. The upper histogram is the number of mis-configured author profiles in the first 10 searching result pages and the lower histogram is that in the first searching result page. The abbreviations of research fields are the same as Table 6.



(**a**) (**b**)

Figure 4. The country distribution of the outstanding scholars. (**a**) The country distribution of all the outstanding scholars. (**b**) The country distribution of outstanding scholars whose author profiles are reliable based on our proposed model.

7.2. Case Study 2: University Ranking

The metrics on the Google Scholar author profiles, such as h-index and the total number of citations, are widely used for evaluating scholars. In this part, we try to evaluate a university based on the scholars working for it. We design the rank criteria and compare the ranking list before and after excluding our detected mis-configured author profiles. The difference between the ranking results reflects the negative impact of mis-configuration on the data analysis, which demonstrates the importance and effectiveness of our detection.

In our dataset, 91.4% of the scholars verified their profiles using institutional email addresses. We utilize the data from Google Scholar to generate university ranking lists and compare these lists with two well-accepted ranking lists, the QS World University Rankings by Subject (Computer Science & Information Systems) [38] and the Times Higher Education World University Ranking table for computer science [39], to evaluate the quality of our ranking lists. Since the dataset is obtained in January 2019, we use the 2018 version of these two ranking lists for comparison. Three metrics are designed to rank universities in this part. The first metric is the h2-index [40] of each university. In detail, since h-index is chosen as the metric to evaluate a single scholar, we use h2-index to aggregate different scholars, which means that there are at least h2 scholars belonging to the university whose h-index values are not less than h2. The second metric is the number of scholars who are highly cited. The definition of a highly cited scholar here is that the total number of citations received by the scholar ranks top 1% among all the scholars in the dataset. The third metric is the number of scholars whose h-index ranks top 1% among all the scholars in the dataset. Since we only study computer science related fields, we do not need to consider the issue of normalization across different fields. The ranking method is relatively simple. We sort universities involved in our dataset by these three metrics, respectively, and obtain the top 20 universities before and after excluding the mis-configured author profiles. Then, we observe their rankings in the two comparative ranking lists.

The ranking result may be a little bit different from the two comparative ranking lists, because our ranking methods are totally based on the publication information from Google Scholar. Although the methods used to rank the universities are different, it can be seen that the selected 20 universities are all ranked relatively high in the two comparative lists especially when we exclude the mis-configured author profiles. In order to quantitatively measure the similarity of different ranking lists, we use two metrics, square deviation value (SDV) and mean deviation value (MDV). The definition of them are as follows:

$$SDV(A,B) = \frac{1}{n}\sum_{i=0}^{n-1}(A[i]-B[i])^2 \tag{16}$$

$$MDV(A,B) = \frac{1}{n}\sum_{i=0}^{n-1}|A[i]-B[i]| \tag{17}$$

where $A[i] = i+1$ represents the the ranking of the university based on our defined metrics. $B[i]$ is the relative ranking of the university in the comparative ranking list. n is the length of ranking list. The smaller SDV and MDV are, the more similar two ranking lists are. The results of these two metrics are shown in Table 7. After excluding the mis-configured author profiles, the resulted top universities rank generally better. In other words, leaving out the mis-configured author profiles can improve the quality of university ranking. The results show that our work to detect mis-configured author profiles is necessary and meaningful. The ranking result based on highly cited scholars is a little bit abnormal. We believe that the reason is that the probability of highly cited scholars having mis-configured author profiles is relatively higher. They usually have a large number of papers in their paper lists, therefore, it may be hard to keep the profiles accurate. Actually, the numbers of citations of these scholars are very large no matter whether their author profiles are mis-configured. So, it may introduce ranking errors when totally filtering out them. A more accurate method is only filtering out the wrongly included papers, and we leave it for the future work.

Table 7. Comparison of different rankings.

Ranking Metrics	Compared Ranking	w/Mis-Configured Profiles		w/o Mis-Configured Profiles	
		SDV	MDV	SDV	MDV
University h2-index	Times Higher Education World University RankingTable for CS	184.8	10.3	175.6	9.3
	QS World University Rankings by Subject(CS & IS)	695.5	18.3	625.6	17.4
Highly cited	Times Higher Education World University RankingTable for CS	541.3	13.3	556.8	13.2
	QS World University Rankings by Subject (CS & IS)	1830.5	23.6	1793.3	23.4
High h-index	Times Higher Education World University RankingTable for CS	539.5	14.5	328.7	11.8
	QS World University Rankings by Subject (CS & IS)	2012.8	27.3	810.8	19.9

8. Limitations

In the paper, we introduce an approach to detect the mis-configured author profiles on Google Scholar. However, there are still some limitations of our work.

Firstly, we believe that mis-configuration means wrongly claiming others' papers in the paper list. However, the author profile owner's missing papers, another scenario of mis-configuration, has not been taken into consideration. It is relatively difficult to detect the incompleteness of the paper list, and we might need to refer to third-party data sources.

Secondly, while crawling data from Google Scholar, we select the research fields from the subdomains in "Metric" page. However, there are sub-sub-domains under these tags, and some computer science related fields are interdisciplinary. Some tags are not covered in this paper, such as Information Theory and IoT. We manually add some tags to the crawling list to expand the coverage of our dataset. However, there are still some tags being omitted, which leads to an incomplete set of computer science authors.

Thirdly, owing to the limitation of labor force and time, we labeled 3000 scholars' author profiles. The labeled data may be biased on certain features.

9. Related Work

This part incorporates some previous work related to ours. First, we list some academic databases besides Google Scholar in Section 9.1. Then, we briefly introduce a research hotspot, name disambiguation, about academic data reliability in Section 9.2. Finally, in Section 9.3, some methods for fake account detection are introduced that can be referred to for the detection of mis-configured author profiles.

9.1. Academic Database

There have been many academic databases of search engines that are widely used in academic data mining.

Microsoft Academic Service (MAS) [41] provides a heterogeneous entity graph consisting of six types of entities: field of study, author, institution (the affiliation of author), paper, venue (journal and conference series) and event. The source of the data is publishers (e.g., ACM and IEEE) and web-pages indexed by Bing. There are two typical functionalities: natural language powered interactive search and heterogeneous recommendation, which take advantage of the relationships across different types of entities.

AMiner [13] is another academic search system providing mining services. It integrates academic data from multiple sources such as DBLP, ACM Digital Library and MAS. A

semantic-based profile is maintained for each researcher. AMiner is widely used in the research about social networks.

CiteSeerX [42] is a scientific literature digital library and search engine focusing on the field of computer and information science. Besides indexing papers, it also provides resources such as metadata, algorithms and software.

9.2. Name Disambiguation

Name ambiguity is a critical issue in digital libraries, which stems from the fact that many scholars share an identical name or a scholar has several distinct names. This issue may affect the performance of scholar information integration and document retrieval. Name disambiguation is usually viewed as a clustering problem and the corresponding solutions can be categorized into two types.

Feature-based methods: The typical approach for name disambiguation is to define similarities between literatures based on features extracted from the literature and then conduct clustering for each person's name based on the similarity. Yoshida et al. [43] proposed a two-stage clustering method to learn feature representation, which only used strong features (e.g., named entities, compound keywords and URLs) in the first stage and leveraged bootstrapping techniques to assign weights to weak features (single words). Khabsa et al. [44] proposed a constraint-based clustering algorithm utilizing DBSCAN with a pairwise distance based on random forests. Louppe et al. [45] performed semi-supervised hierarchical clustering of publications by distinct authors.

Linkage-based methods: Another type of method for name disambiguation is to leverage the information extracted from the academic social network. GHOST (GrapH-based framewOrk for name diStincTion) [46] built a co-author graph in which authors sharing an ambiguous name are treated as different nodes (document graph for each name). Furthermore, they exploited the relationship between every pair of papers. A clustering algorithm called Affinity Propagation (AP) algorithm was used to generate clustering results. Zhang et al. [47] designed a representation learning model to obtain the low-dimensional document embedding from graphs based on document similarity and co-author relationship. Name disambiguation can be solved using a hierarchical agglomerative clustering algorithm. Tang et al. [13] proposed the ArnetMiner system to extract and mine academic social networks. A probabilistic framework is proposed to deal with the name ambiguity problem. Hidden Markov Random Fields were utilized by Tang et al. [48] to incorporate node features and edge features in a unified probabilistic framework. Zhang et al. [49] proposed a unified framework, which leverages global metric and local linkage to estimate the number of people sharing the same name and deal with the name ambiguity problem. Particularly, they involved human effort to improve the accuracy.

9.3. Fake Account Detection

Fake accounts can severely affect the social activities in online social networks (OSNs). A malicious user may create multiple fake accounts to perform cybercrimes, such as rigging popularity, scamming and breaking trust in online associations. Therefore, it is very important to detect fake accounts to eliminate their negative impact. In general, besides manual verification, the schemes to automatically detect fake accounts can be divided into two categories: graph-based schemes and machine learning-based schemes.

Graph-based schemes: Graph-based schemes enable fake account detection by analyzing the social graph. SybilGuard [50] and SybilLimit [51] were based on the assumption that there are two kinds of regions in a social network. A honest region consists of honest nodes, while a sybil region is composed of sybil nodes created by malicious users. SybilShield [52] assumed that a network was composed of small, medium and large communities. They employed random walks to identify fake accounts. SybilInfer [53] leveraged a Bayesian inference approach for fake account defense. SybilRank [54] ranked users based on their perceived likelihood to be fake accounts. Gong et al. [55] proposed a semi-supervised

method called SybilBelief, which relied on loopy belief propagation and Markov random fields. In conclusion, graph-based methods leverage the social connections between users to uncover fake accounts.

Machine learning-based schemes: Machine learning is used for autonomously acquiring knowledge from observations involving supervised learning, unsupervised learning and semi-supervised learning. Yang et al. [56] compared the performances of a simple threshold detector and a SVM detector on Renren. Wang et al. [57] analyzed users' clickstream activity and detected fake accounts using clustering. DeepScan [58] exploited LSTM to extract user dynamic behavior and identified malicious accounts. DatingSec [59] leveraged LSTM and an attentive module to capture the interplay of users' temporal-spatial behaviors and user-generated textual content for malicious account detection in dating apps. SybilExposer [60] was a fake account community detection algorithm, which used the properties of social graph communities to rank communities according to their likelihood to be fake.

10. Conclusions and Future Work

In this paper, we focused on the detection of mis-configured author profiles on Google Scholar. We obtained over 170 thousand author profiles using a distributed crawler and manually labeled 3000 of them by a crowdsourcing-based labeling system as the ground truth. We proposed an integrated model to automatically detect the mis-configured author profiles that leveraged neural network, node embedding and machine learning techniques. The F1-score of our model reached 0.9610, which outperformed other baselines. Finally, two application case studies were conducted based on the data before and after filtering out mis-configured author profiles. The results validate the importance and meaningfulness of our work.

In the future, we would recruit more students to label more scholars and alleviate the ratio of labeling error and improve the prediction accuracy. Additionally, semi-supervised learning [61] is a popular method to deal with the challenge of the small number of labeled data, which can be a research direction we will resort to.

Author Contributions: Conceptualization, J.T. and Y.C.; methodology, J.T. and G.S.; validation, J.T.; formal analysis, J.T.; data curation, J.T. and G.S.; writing—original draft preparation, J.T. and Y.C.; writing—review and editing, J.T., Y.C., Y.X., K.S., X.W., Y.W., Z.Z. and P.H.; funding acquisition, Y.C. and X.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Natural Science Foundation of China (No. 62072115, No. 71731004, No. 61602122, No. 61971145). This research was also supported by Meituan.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The author profile dataset used is available at figshare: www.doi.org/10.6084/m9.figshare.14208380 (accessed on 1 June 2021). The labeled author profile dataset is available at figshare: www.doi.org/10.6084/m9.figshare.14208332 (accessed on 1 June 2021).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Katz, G.; Rokach, L. Wikiometrics: A Wikipedia based ranking system. *World Wide Web* **2017**, *20*, 1153–1177. [CrossRef]
2. Hug, S.E.; Ochsner, M.; Brändle, M.P. Citation analysis with microsoft academic. *Scientometrics* **2017**, *111*, 371–378. [CrossRef]
3. He, Y.; Hui, S.C. Mining a Web Citation Database for author co-citation analysis. *Inf. Process. Manag.* **2002**, *38*, 491–508. [CrossRef]
4. Gusenbauer, M. Google Scholar to overshadow them all? Comparing the sizes of 12 academic search engines and bibliographic databases. *Scientometrics* **2019**, *118*, 177–214. [CrossRef]
5. Chen, Y.; Ding, C.; Hu, J.; Chen, R.; Hui, P.; Fu, X. Building and Analyzing a Global Co-Authorship Network Using Google Scholar Data. In Proceedings of the 26th International Conference on World Wide Web Companion, Perth, Australia, 3–7 April 2017; pp. 1219–1224.
6. Falagas, M.E.; Pitsouni, E.I.; Malietzis, G.A.; Pappas, G. Comparison of PubMed, Scopus, Web of Science, and Google Scholar: strengths and weaknesses. *FASEB J.* **2008**, *22*, 338–342. [CrossRef]

7. Li, J.; Burnham, J.F.; Lemley, T.; Britton, R.M. Citation analysis: Comparison of Web of Science®, Scopus™, SciFinder®, and Google Scholar. *J. Electron. Resour. Med. Libr.* **2010**, *7*, 196–217. [CrossRef]
8. Schmalensee, R. Using the H-index of concentration with published data. *Rev. Econ. Stat.* **1977**, *59*, 186–193. [CrossRef]
9. Mingers, J.; O'Hanley, J.R.; Okunola, M. Using Google Scholar institutional level data to evaluate the quality of university research. *Scientometrics* **2017**, *113*, 1627–1643. [CrossRef]
10. Smalheiser, N.R.; Torvik, V.I. Author Name Disambiguation. *Annu. Rev. Inf. Sci. Technol.* **2009**, *43*, 1–43. [CrossRef]
11. Han, H.; Giles, L.; Zha, H.; Li, C.; Tsioutsiouliklis, K. Two supervised learning approaches for name disambiguation in author citations. In Proceedings of the 4th Joint ACM/IEEE Conference on Digital Libraries, Tucson, AZ, USA, 11 June 2004; pp. 296–305.
12. Torvik, V.I.; Smalheiser, N.R. Author Name Disambiguation in MEDLINE. *ACM Trans. Knowl. Discov. Data* **2009**, *3*, 11:1–11:29. [CrossRef] [PubMed]
13. Tang, J.; Zhang, J.; Yao, L.; Li, J.; Zhang, L.; Su, Z. ArnetMiner: Extraction and Mining of Academic Social Networks. In Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Las Vegas, NV, USA, 24–27 August 2008; pp. 990–998.
14. Nigam, A.; Nigam, P.K. Citation index and impact factor. *Indian J. Dermatol. Venereol. Leprol.* **2012**, *78*, 511–516. [CrossRef]
15. Cloud Translation/Google Cloud. Available online: https://cloud.google.com/translate (accessed on 1 June 2021).
16. Blei, D.M.; Ng, A.Y.; Jordan, M.I. Latent Dirichlet Allocation. *J. Mach. Learn. Res.* **2003**, *3*, 993–1022.
17. Lee, D.D.; Seung, H.S. Learning the parts of objects by non-negative matrix factorization. *Nature* **1999**, *401*, 788. [CrossRef]
18. Rousseeuw, P.J. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.* **1987**, *20*, 53–65. [CrossRef]
19. Welch, B.L. The Generalization of 'Student's' Problem when Several Different Population Variances are Involved. *Biometrika* **1947**, *34*, 28–35. [CrossRef] [PubMed]
20. Kent, J.T. Information gain and a general measure of correlation. *Biometrika* **1983**, *70*, 163–173. [CrossRef]
21. Kim, Y. Convolutional Neural Networks for Sentence Classification. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, Doha, Qatar, 25–29 October 2014.
22. Koutník, J.; Greff, K.; Gomez, F.; Schmidhuber, J. A Clockwork RNN. In Proceedings of the 31st International Conference on Machine Learning, Beijing, China, 21–26 June 2014; Volume 32, pp. II-1863–II-1871.
23. Grover, A.; Leskovec, J. Node2Vec: Scalable Feature Learning for Networks. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 855–864.
24. Wang, W.; Xu, B.; Liu, J.; Cui, Z.; Yu, S.; Kong, X.; Xia, F. CSTeller: Forecasting scientific collaboration sustainability based on extreme gradient boosting. *World Wide Web* **2019**, *22*, 2749–2770. [CrossRef]
25. Ke, G.; Meng, Q.; Finley, T.; Wang, T.; Chen, W.; Ma, W.; Ye, Q.; Liu, T.Y. LightGBM: A Highly Efficient Gradient Boosting Decision Tree. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; Volume 30, pp. 3146–3154.
26. Li, M.; Zhang, T.; Chen, Y.; Smola, A.J. Efficient Mini-batch Training for Stochastic Optimization. In Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, NY, USA, 24–27 August 2014; pp. 661–670.
27. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.
28. Fawcett, T. An introduction to ROC analysis. *Pattern Recognit. Lett.* **2006**, *27*, 861–874. [CrossRef]
29. Cox, D.R. The Regression Analysis of Binary Sequences. *J. R. Stat. Soc. Ser. (Methodol.)* **1958**, *20*, 215–232. [CrossRef]
30. Safavian, S.; Landgrebe, D. A survey of decision tree classifier methodology. *IEEE Trans. Syst. Man Cybern.* **1991**, *21*, 660–674. [CrossRef]
31. Breiman, L. Random Forests. *Mach. Learn.* **2001**, *45*, 5–32. [CrossRef]
32. Wang, B.; Jia, J.; Zhang, L.; Gong, N.Z. Structure-Based Sybil Detection in Social Networks via Local Rule-Based Propagation. *IEEE Trans. Netw. Sci. Eng.* **2019**, *6*, 523–537. [CrossRef]
33. Yang, Z.; Yang, D.; Dyer, C.; He, X.; Smola, A.; Hovy, E. Hierarchical attention networks for document classification. In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego, CA, USA, 12–17 June 2016; pp. 1480–1489.
34. McNemar, Q. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika* **1947**, *12*, 153–157. [CrossRef]
35. Gers, F.A.; Schmidhuber, J.; Cummins, F. Learning to Forget: Continual Prediction with LSTM. *Neural Comput.* **2000**, *12*, 2451–2471. [CrossRef] [PubMed]
36. Chang, S.; Zhang, Y.; Han, W.; Yu, M.; Guo, X.; Tan, W.; Cui, X.; Witbrock, M.; Hasegawa-Johnson, M.A.; Huang, T.S. Dilated Recurrent Neural Networks. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; Volume 30, pp. 77–87.
37. MaxMind GeoLite Databases. Available online: https://dev.maxmind.com/geoip/geoip2/geolite2/ (accessed on 1 June 2021).
38. Computer Science & Information Systems/Top Universities. Available online: https://www.topuniversities.com/university-rankings/university-subject-rankings/2018/computer-science-information-systems (accessed on 1 June 2021).

39. World University Rankings 2019 by Subject: Computer Science/Times Higher Education (THE). Available online: https://www.timeshighereducation.com/world-university-rankings/2018/subject-ranking/computer-science (accessed on 1 June 2021).
40. Schubert, A. Successive H-Indices. *Scientometrics* **2007**, *70*, 201–205. [CrossRef]
41. Sinha, A.; Shen, Z.; Song, Y.; Ma, H.; Eide, D.; Hsu, B.J.P.; Wang, K. An Overview of Microsoft Academic Service (MAS) and Applications. In Proceedings of the 24th International Conference on World Wide Web, Florence, Italy, 18–22 May 2015; pp. 243–246.
42. Li, H.; Councill, I.; Lee, W.C.; Giles, C.L. CiteSeerx: An Architecture and Web Service Design for an Academic Document Search Engine. In Proceedings of the 15th International Conference on World Wide Web, Scotland, UK, 23–26 May 2006; pp. 883–884.
43. Yoshida, M.; Ikeda, M.; Ono, S.; Sato, I.; Nakagawa, H. Person Name Disambiguation by Bootstrapping. In Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval, Geneva, Switzerland, 19–23 July 2010; pp. 10–17.
44. Khabsa, M.; Treeratpituk, P.; Giles, C.L. Online Person Name Disambiguation with Constraints. In Proceedings of the 15th ACM/IEEE-CS Joint Conference on Digital Libraries, Knoxville, TN, USA, 21–25 June 2015; pp. 37–46.
45. Louppe, G.; Al-Natsheh, H.T.; Susik, M.; Maguire, E.J. Ethnicity Sensitive Author Disambiguation Using Semi-supervised Learning. In Proceedings of the International Conference on Knowledge Engineering and the Semantic Web, Prague, Czech Republic, 21–23 September 2016; pp. 272–287.
46. Fan, X.; Wang, J.; Lv, B.; Zhou, L.; Hu, W. GHOST: An Effective Graph-based Framework for Name Distinction. In Proceedings of the 17th ACM Conference on Information and Knowledge Management, Napa Valley, CA, USA, 26–30 October 2008; pp. 1449–1450.
47. Zhang, B.; Al Hasan, M. Name Disambiguation in Anonymized Graphs Using Network Embedding. In Proceedings of the 2017 ACM Conference on Information and Knowledge Management, Singapore, 6–10 November 2017; pp. 1239–1248.
48. Tang, J.; Fong, A.C.M.; Wang, B.; Zhang, J. A Unified Probabilistic Framework for Name Disambiguation in Digital Library. *IEEE Trans. Knowl. Data Eng.* **2012**, *24*, 975–987. [CrossRef]
49. Zhang, Y.; Zhang, F.; Yao, P.; Tang, J. Name Disambiguation in AMiner: Clustering, Maintenance, and Human in the Loop. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, London, UK, 19–23 August 2018; pp. 1002–1011.
50. Yu, H.; Kaminsky, M.; Gibbons, P.B.; Flaxman, A. SybilGuard: Defending against Sybil Attacks via Social Networks. In Proceedings of the 2006 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, Pisa, Italy, 11–15 September 2006; pp. 267–278.
51. Yu, H.; Gibbons, P.B.; Kaminsky, M.; Xiao, F. Sybillimit: A Near-Optimal Social Network Defense against Sybil Attacks. In Proceedings of the IEEE Symposium on Security and Privacy, Oakland, CA, USA, 21–23 May 2008; pp. 3–17.
52. Shi, L.; Yu, S.; Lou, W.; Hou, Y.T. Sybilshield: An agent-aided social network-based Sybil defense among multiple communities. In Proceedings of the IEEE International Conference on Computer Communications, Nassau, Bahamas, 30 July–2 August 2013; pp. 1034–1042.
53. Danezis, G.; Mittal, P. SybilInfer: Detecting Sybil Nodes using Social Networks. In Proceedings of the Network and Distributed System Security Symposium, Diego, CA, USA, 8–11 February 2009; pp. 1–15.
54. Cao, Q.; Sirivianos, M.; Yang, X.; Pregueiro, T. Aiding the Detection of Fake Accounts in Large Scale Social Online Services. In Proceedings of the 9th Symposium on Networked Systems Design and Implementation, San Jose, CA, USA, 25–27 April 2012; pp. 197–210.
55. Gong, N.Z.; Frank, M.; Mittal, P. Sybilbelief: A Semi-Supervised Learning Approach for Structure-Based Sybil Detection. *IEEE Trans. Inf. Forensics Secur.* **2014**, *9*, 976–987. [CrossRef]
56. Yang, Z.; Wilson, C.; Wang, X.; Gao, T.; Zhao, B.Y.; Dai, Y. Uncovering Social Network Sybils in the Wild. *ACM Trans. Knowl. Discov. Data* **2014**, *8*, 2:1–2:29. [CrossRef]
57. Mukherjee, A.; Kumar, A.; Liu, B.; Wang, J.; Hsu, M.; Castellanos, M.; Ghosh, R. Spotting Opinion Spammers Using Behavioral Footprints. In Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Chicago, IL, USA, 11–14 August 2013; pp. 632–640.
58. Gong, Q.; Chen, Y.; He, X.; Zhuang, Z.; Wang, T.; Huang, H.; Wang, X.; Fu, X. DeepScan: Exploiting Deep Learning for Malicious Account Detection in Location-Based Social Networks. *IEEE Commun. Mag.* **2018**, *56*, 21–27. [CrossRef]
59. He, X.; Gong, Q.; Chen, Y.; Zhang, Y.; Wang, X.; Fu, X. DatingSec: Detecting Malicious Accounts in Dating Apps Using a Content-Based Attention Network. *IEEE Trans. Dependable Secur. Comput.* **2021**. [CrossRef]
60. Misra, S.; Tayeen, A.S.M.; Xu, W. SybilExposer: An effective scheme to detect Sybil communities in online social networks. In Proceedings of the IEEE International Conference on Communications, Kuala Lumpur, Malaysia, 23–27 May 2016; pp. 1–6.
61. Board, R.; Pitt, L. Semi-supervised learning. *Mach. Learn.* **1989**, *4*, 41–65. [CrossRef]